III. REMARKS

In this Amendment, claim 1 has been amended. Claims 2-18 remain unchanged. In sum, claims 1-18 are pending after entry of this Amendment.

A. Rejection of Claims 1-9 Under 35 U.S.C. §101

Claims 1-9 have been rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. The Examiner asserts that claims 1-9 are directed to method steps which can be practiced mentally in conjunction with pen and paper, and therefore they are directed to non-statutory subject matter. Specifically, as claimed, it is uncertain what performs each of the claimed method steps.

As suggested by the Examiner, claim 1 has been amended to recite "computer implemented method" to overcome this rejection. Claims 2-9 depend from claim 1 and accordingly incorporate this limitation by reference.

Accordingly, Applicants respectfully request that the instant rejection of claims 1-9 be withdrawn.

B. Rejection Of Claims 1, 10, And 18 Under 35 U.S.C. §102(b) As Being Anticipated By Microstrategy

Claims 1, 10, and 18 were rejected under 35 U.S.C. §102(b) as being anticipated by "MicroStrategy Administrator" Version 6.0 (hereinafter "MicroStrategy") published in October, 1999. This reference was cited in an Information Disclosure Statement submitted on September 30, 2002.

Microstrategy's Introduction states that "[MicroStrategy Object Manager] provides topographical viewing and manipulation of all decision support project and user information on one screen, and enables projects to be transferred safely and easily from testing to production

environments." MicroStrategy, page vii. In other words, the MicroStrategy software product enables human users to view and manipulate projects.

In contrast, amended claim 1 and original claims 10 and 18 recite various functions that are accomplished either "automatically" (i.e., without human intervention) (claim 1), "using a computer processor" (claim 1), by an "operational module" (claim 10), or via "code" on a "processor-readable medium" (claim 18). These functions are accomplished either without human intervention or by a computer processor.

Claim 1 recites "determining using a computer processor an appropriate manner of executing the selected function." The Office Action asserts that this limitation is taught by MicroStrategy's instructions regarding copying between users: "Click Copy on the dialog bar. Then highlight a selection and click Paste on the dialog bar to copy over the selection." Office Action, p. 3 (citing MicroStrategy, p. 28, ¶ 4). The asserted passage of MicroStrategy relates to an action (copying) that requires human intervention (e.g., to click copy on the dialog bar). Furthermore, the manner of performing this action (by clicking copy and highlighting) was not determined using a computer processor or without human intervention, as required by the claims. Rather, these instructions were written by humans as a guide for copying between users. This manner of copying was not determined by a computer processor.

For at least these reasons, Applicants respectfully request that the instant rejection of claims 1, 10, and 18 be withdrawn.

C. Rejection Of Claims 1, 3-5, 10-12 And 16-18 Under 35 U.S.C. §102(b) As Being Anticipated By Mariani.

Claims 1, 3-5, 10-12, and 16-18 were rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,854,932 to Mariani et al. ("Mariani").

Mariani teaches "[a] minimal rebuild system and process for minimizing rebuilding of a user's programming project analyzes and records dependencies of object code files compiled in a previous build of the project on classes declared in header files. When rebuilding the project, the system detects and records changes made to the classes and header files since the project was last built. The system then determines whether to recompile the object code files from their respective source code files by comparing the recorded dependencies and changes. If an object code file's dependencies do not intersect the changes, recompiling of the object code file can be omitted. The minimal rebuild system utilizes an approximate representation of the dependencies to yield an efficient system while ensuring that the project is correctly rebuilt." Mariani Abstract.

The Office Action asserts that Mariani's system for selectively compiling object code teaches the claims. Specifically, the Office Action asserts the following correlations between the elements of amended claim 1 and the Mariani system. As asserted in the Office Action, the claims 10 and 18 have similar limitations to these limitations (i) through (vi).

> (i) "receiving a command to perform a selected function on a selected object" corresponds to users directly modifying source code files and header files;
>
> (ii) "automatically identifying dependent objects referred to by the selected object" refers to generating dependency information for an object code file;
>
> (iii) "determining using a computer processor an appropriate manner of executing the selected function on the selected object" corresponds to determining when recompiling can be avoided by determining how the object code files are dependent on the header files;
>
> (iv) "determining using a computer processor appropriate functions to be performed on the dependent objects" corresponds to selectively recompiling the source code files so as to detect changes to all header files that were changed since the last project build, wherein the system then utilizes the detected changes to the header files to determine which of the remaining source code files to recompile and which can be avoided;
>
> (v) "automatically causing the appropriate functions to be performed on the dependent objects" corresponds to the act of either compiling a selected source file into an object file or, if the compiling was omitted for that source file, resaving the object file; and

(vi) "automatically causing the execution of the selected function on the selected object in the appropriate manner" corresponds to recompiling the source code files that were changed since the last project build.

Office Action, pp. 5-6 (amendments added).

First, Mariani does not teach or suggest "determining using a computer processor an appropriate **manner** of executing the selected function" as recited in limitation (iii) of claim 1. The Office Action asserts that determining when a function (e.g., recompiling) can be avoided (or not avoided) amounts to determining an "appropriate manner" of performing the function. The word "manner" refers to the way that something is done, not the fact of whether or not it is done. Similarly, the limitation "determining whether to execute a selected function" is not an example of "determining...an appropriate **manner** of executing the selected function," as recited in claim 1. To assert that a "manner" can be taught by a simple "yes or no" determination strips the word "manner" of any meaning. To put it another way, an "appropriate manner" of performing a function cannot be to perform the function, nor can it be to not perform the function.

Further, the cited features of Mariani do not collectively teach the elements of claim 1 because they fail to maintain a consistent notion of the recited claim elements (in particular, the recited nouns such as "selected object"). In other words, the Office Action has not shown a single, coherent embodiment of Mariani that teaches all the elements of the claims. Rather, the Office Action applies disparate features of Mariani that accomplish dissociated tasks. For instance, to teach the "selected function" of claim 1, the Office Action alternately applies "modifying code" in element (i) and "recompiling" in elements (iii) and (vi). Similarly, to teach a "selected object," the Office Action appears to apply a "source code file" in element (i), an "object code file" in element (ii), and specific source code files that were changed since the last project build in element (vi).

Thus, Mariani fails to teach claim 1 as well as claims 10 and 18, which have some related limitations. The same arguments apply to claims 3-5, 11, 12, 16, and 17, which depend from and incorporate the limitations of claims 1 and 10.

For at least these reasons, Applicants respectfully request that the instant rejection of claims 1, 3-5, 10-12, and 16-18 be withdrawn.

D. Rejection Of Claims 2, 6-9 And 13-15 Under 35 U.S.C. §103(a) As Being Unpatentable Over Mariani in view of U.S. Patent No. 6,112,024 to Almond.

Claims 2, 6-9 And 13-15 were rejected under 35 U.S.C. §103(a) as being unpatentable over Mariani in view of U.S. Patent No. 6,112,024 to Almond et al. ("Almond").

Mariani does not teach or suggest the elements of claims 1 and 10 as discussed above. Because claims 2, 6-9, and 13-15 depend from claims 1 and 10 and therefore incorporate the limitations of claims 1 and 10 by reference, Mariani similarly fails to teach or suggest claims 2, 6-9, and 13-15. Almond fails to remedy the deficiencies of Mariani. Thus, the combination of Mariani and Almond fails to teach or suggest the elements of claim 2, 6-9, and 13-15.

As for claim 6, the Office Action asserts that Mariani as modified teaches the limitation of receiving a command to copy a selected object from a source project to a destination project. In particular, the Office Action cites a passage of Almond stating "the user will perform versioning activities such as...Get -- copy one or multiple objects to the user's local directory." Almond, col. 39, lines 28-32. Almond's reference to copying objects fails to teach the subject matter of claim 6 because this feature cannot be combined with Mariani in the manner prescribed by claim 6.

Claim 6 recites "wherein the step of receiving is a step of receiving a command to copy a selected object from a source project to a destination project." Claim 6 also incorporates the limitations of claim 1, which in the context of claim 6 require the "identifying,"

"determining...manner," "determining...functions," "causing...functions," and "causing the execution" actions to be performed in accordance with the "command to copy a selected object from a source project to a destination project." Mariani teaches away from a combination with Almond in this context. The Office Action uses Mariani's system for selectively recompiling object code to purportedly establish how Mariani teaches the elements of claim 1. However, while the Mariani system is specifically designed to selectively recompile object code, it is not configured to copy objects in the manner by which it recompiles object code. Copying and recompiling are completely different operations that require completely different systems and methods. Thus even if Mariani taught the limitations of claim 1 (which it does not), the Mariani system cannot be combined with Almond to copy objects in the way that it selectively recompiles object code. Thus, the combination of Mariani and Almond does not teach or suggest claim 6.

For at least these reasons, Applicants respectfully request that the instant rejection of claims 2, 6-9 and 13-15 be withdrawn.

IV. CONCLUSION

For all the reasons set forth above, it is respectfully submitted that all outstanding objections and rejections have been overcome or rendered moot. Further, all pending claims are patentably distinguishable over the prior art of record. Any amendments are supported by the specification. Applicants accordingly submit that these claims are in a condition for allowance. Reconsideration and allowance of all claims are respectfully requested.

Authorization is hereby granted to charge or credit the undersigned's Deposit Account No. 50-0206 for any fees or overpayments related to the entry of this Amendment, including any extension of time fees and new claims fees.

Respectfully submitted,

HUNTON & WILLIAMS LLP

Dated: 11/8/04

By: Thomas D. Bradshaw

Thomas D. Bradshaw
Registration No. 51,492

for Brian Buroker
Registration No. 39,125

HUNTON & WILLIAMS LLP
Intellectual Property Department
1900 K Street, NW, Suite 1200
Washington, DC 20006-1109
(202) 955-1500 (Telephone)
(202) 778-2201 (Facsimile)